[Yang, et al., 2015], the robots' waiting time is set as $\omega_{20}=6$ s, $\omega_{21}=2$ s, and $\omega_{22}=\Pi-\psi_{21}-\omega_{20}-\omega_{21}=100-30-6-2=62$ s; $\omega_{30}=2$ s, $\omega_{31}=4$ s, and $\omega_{32}=\Pi-\psi_{31}-\omega_{30}-\omega_{31}=100-42-2-4=52$ s. Thus, for C_1, as $\Pi-(4\lambda_2+3\mu_2+\omega_{22})-\lambda_1=100-(19+62)-23=-4<0$ and $\Pi-(4\lambda_3+3\mu_3+\omega_{32})-\lambda_1=100-(27+52)-23=-2<0$, or (14) is violated and there is no OSLB. Thus one needs to find an optimal cycle time Θ by Algorithm 2.

[0194] By Algorithm 2, one has $\Delta_2=\Omega_2(D, S)/n[2]=4/2=2$ s, $\Delta_3=\Omega_3(D, S)/n[3]=2/2=1$ s, $\Delta=\max\{\Delta_2, \Delta_3\}=2$ s, and $\Theta=102$ s. Then, let $A_{ij}=\omega_{ij}$, $i\in\mathbb{N}_3\backslash\{1\}$ and $j\in\Omega_{i(n[i])}$, where ω_{ij} is obtained by the algorithm provided in [Yang, et al., 2015]. By Algorithm 2, the robot waiting time is set as $\omega_{30}=A_{30}+\Delta=4$ s, $\omega_{31}=A_{31}+\Delta=6$ s, $\omega_{32}=A_{32}-\Delta=50$ s, $\omega_{20}=A_{20}+\Delta=8$ s, $\omega_{21}=A_{21}+\Delta=4$ s, $\omega_{22}=A_{22}-\Delta=60$ s, $\omega_{10}=A_{10}+\Delta=6$ s, and $\omega_{11}=\omega_{12}=\omega_{13}=0$. In this way, the minimal cycle time and optimal one-wafer cyclic schedule is obtained and it is shown by the Gantt chart in FIG. **6**.

Example 2

[0195] It is from [Yang, et al., 2015]. A treelike hybrid 5-cluster tool with C_2 be a fork tool, and its adjacent downstream tools are C_3 and C_5. The tool C_4 is the downstream tool of C_3, and C_2 is the downstream tool of C_1. Furthermore, C_i is a dual-arm tool and the others are single-arm tools. Their activity time is as follows: for C_1, one has $(\alpha_{10}, \alpha_{11}, \lambda_1, \mu_1)=(0, 61.5, 0, 28.5, 0.5)$; for C_2, one has $(\alpha_{20}, \alpha_{21}, \alpha_{22}, \lambda_2, \mu_2)=(0, 0, 0, 10, 1)$; for C_3, one has $(\alpha_{30}, \alpha_{31}, \alpha_{32}, \alpha_{33}, \lambda_3, \mu_3)=(0, 56, 0, 58, 7, 1)$; for C_4, one has $(\alpha_{40}, \alpha_{41}, \alpha_{42}, \alpha_{43}, \lambda_4, \mu_4)=(0, 56, 66, 65, 5, 1)$; and for C_5, one has $(\alpha_{50}, \alpha_{51}, \alpha_{52}, \lambda_5, \mu_5)=(0, 48, 50, 6, 1)$.

[0196] From [Yang, et al., 2015], the lower bound of cycle time of the system is $\Theta=\Pi=90$ s. With $\Theta=\Pi=90$ s, the robot waiting time is set as follows. For C_4, $\omega_{40}=11$ s, $\omega_{41}=1$ s, $\omega_{42}=2$ s, and $\omega_{43}=28$ s are set. For C_5, $\omega_{50}=15$ s, $\omega_{51}=13$ s, and $\omega_{52}=20$ s are set. For C_3, $\omega_{31}=8$ s, $\omega_{30}=3$ s, $\omega_{32}=1$ s, and $\omega_{33}=14$ s are set. For C_2, $\omega_{20}=2$ s, $\omega_{21}=0$, and $\omega_{22}=22$ s are set. Then, for C_1, one has $\Pi-(4\lambda_2+3\mu_2+\omega_{22})-\lambda_1=90-(43+22)-28.5=-3.5<0$, or there is no OSLB. Therefore, one needs to find the minimal cycle time Θ by Algorithm 2.

[0197] With $\omega_{22}=22>0$, $\omega_{52}=20>0$, $\omega_{33}=14>0$, and $\omega_{43}=28>0$, one has $\Sigma_{p\in S_2}B[p]=(B[2]+B[3]+B[4]+B[5])=(1+2+2+1)=6$ and $\Omega_2(D, 5)=\Phi_2(D,S)=(4\lambda_2+3\mu_2+\omega_{22})+\lambda_1-\Pi=3.5$ s. Then, $\Delta_2=\Omega_2(D, S)/(\Sigma_{p\in S_2}B[p]+1)=3.5/7=0.5$ s, $\Delta_3=\omega_{33}/(B[3]\ B[4])=14/3$ s, $\Delta_4=\omega_{43}/B[4]=28/2=14$ s, $\Delta_5=\omega_{52}/B[5]=20/1=20$ s, $\Delta=\min\{\Delta_2, \Delta_3, \Delta_4, \Delta_5\}=0.5$ s, and $\Theta=\Pi+\Delta=90.5$ s. Let $\Delta_{ij}=\omega_{ij}$, $i\in\mathbb{N}_5\backslash\{1\}$ and $j\in\Omega_{i(n[i])}$, where ω_{ij}'s are obtained by the algorithm in [Yang, et al., 2015] with cycle time $\Theta=\Pi=90$ s as given above. Then, by Algorithm 2, the robot waiting time is set as follows. For C_4, $\omega_{40}=A_{40}+\Delta=11.5$ s, $\omega_{41}=A_{41}+\Delta=1.5$ s, $\omega_{42}=A_{42}+\Delta-2.5$ s, and $\omega_{43}=A_{43}-2\Delta-27$ s; for C_3, $\omega_{31}=A_{31}+2\Delta+\Delta=9.5$ s, $\omega_{30}=A_{30}+\Delta=3.5$ s, $\omega_{32}=A_{32}+\Delta=1.5$ s, and $\omega_{33}=A_{33}-2\Delta-2\Delta=12$ s; for C_5, $\omega_{50}=A_{50}+\Delta-15.5$ s, $\omega_{51}=A_{51}+\Delta=13.5$ s, and $\omega_{52}=A_{52}+\Delta=19.5$ s; for C_2, $\Omega_{20}=A_{20}+4\Delta+\Delta=4.5$ s, $\omega_{21}=A_{21}+\Delta+\Delta=1$ s, and $\omega_{22}=A_{22}-4\Delta-\Delta-\Delta=19$ s; and for C_1, $\omega_{10}=\Theta-\psi_{11}=3.5$ s and $\omega_{il}=\omega_{12}=0$. In this way, an optimal one-wafer cyclic schedule is obtained and its Gantt chart is shown in FIG. **7**.

E. The Present Invention

[0198] The present invention is developed based on the theoretical development in Sections A-C above.

[0199] An aspect of the present invention is to provide a computer-implemented method for scheduling a treelike hybrid K-cluster tool to generate a one-wafer cyclic schedule. The K-cluster tool has K single-cluster tools.

[0200] The method comprises given a value of cycle time, generating a part of the schedule for a section of the K-cluster tool by performing a generating algorithm. The section of the K-cluster tool is either an EST or a ST. The generating algorithm is based on Algorithm 1 above. In particular, the generating algorithm for EST_k or ST_k, with C_i being a downstream adjacent tool of C_k and with Θ being the given value of cycle time for EST_i, ST_i or B_i comprises Step 3 of Algorithm 1 under a condition that the checking result of Step 1 is negative. Preferably, the generating algorithm further comprises Step 2 and Step 4 of Algorithm 1, provided that the checking result of Step 1 is negative.

[0201] The method is further elaborated based on Algorithm 2 as follows. First identify ST_j, with $j=\max_{l\in F}\{l\}$, and one or more ESTs of ST_j in the K-cluster tool. The one or more ESTs are denoted as EST_{j-1}, EST_{j-2} down to EST_i such that an upstream adjacent tool of C_i is a fork tool. A first part of the schedule for ST_j is first determined by performing the generating algorithm. Then determine a second part of the schedule for EST_{j-1} based on the first part of the schedule. Determining one part of the schedule EST_{j-m} based on a determined part of the schedule for EST_{j-m+1} is repeated until the one or more ESTs are scheduled.

[0202] The embodiments disclosed herein may be implemented using general purpose or specialized computing devices, computer processors, or electronic circuitries including but not limited to digital signal processors (DSP), application specific integrated circuits (ASIC), field programmable gate arrays (FPGA), and other programmable logic devices configured or programmed according to the teachings of the present disclosure. Computer instructions or software codes running in the general purpose or specialized computing devices, computer processors, or programmable logic devices can readily be prepared by practitioners skilled in the software or electronic art based on the teachings of the present disclosure.

[0203] In particular, the method disclosed herein can be implemented in a treelike hybrid K-cluster tool if the K-cluster tool includes one or more processors. The one or more processors are configured to execute a process of scheduling the K-cluster tool according to one of the embodiments of the disclosed method.

[0204] The present invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiment is therefore to be considered in all respects as illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than by the foregoing description, and all changes that come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

What is claimed is:

1. A computer-implemented method for scheduling a treelike hybrid K-cluster tool to generate a one-wafer cyclic schedule, the treelike hybrid K-cluster tool having K single-cluster tools denoted as $C_1, C_2, \ldots, C_K$, with C_1 being a head tool of the treelike hybrid K-cluster tool, the single-cluster tool C_k, $k\in\mathbb{N}_K$, having a robot R_k for wafer handling, the method comprising: